
feedinlib

Release 0.0.0

Aug 22, 2022

Contents

| | | |
|-----------|---|-----------|
| 1 | Overview | 1 |
| 1.1 | Installation | 1 |
| 1.2 | Documentation | 1 |
| 1.3 | Development | 1 |
| 2 | Getting started | 3 |
| 2.1 | Introduction | 3 |
| 2.2 | Installation | 3 |
| 2.3 | Examples and basic usage | 3 |
| 2.4 | Contributing | 4 |
| 2.5 | Citing the feedinlib | 4 |
| 2.6 | License | 5 |
| 3 | Installation | 7 |
| 4 | Usage | 9 |
| 5 | Examples | 11 |
| 6 | API | 13 |
| 6.1 | Power plant classes | 13 |
| 6.2 | Feed-in models | 13 |
| 6.3 | Weather data | 13 |
| 6.4 | Tools | 14 |
| 6.5 | Abstract classes | 14 |
| 7 | Reference | 15 |
| 7.1 | feedinlib | 15 |
| 8 | Contributing | 17 |
| 8.1 | Bug reports | 17 |
| 8.2 | Documentation improvements | 17 |
| 8.3 | Feature requests and feedback | 17 |
| 8.4 | Development | 18 |
| 9 | Authors | 19 |
| 10 | Changelog | 21 |

| | | |
|-----------|---------------------------------------|-----------|
| 10.1 | 0.0.0 (2021-06-10) | 21 |
| 11 | What's New | 23 |
| 11.1 | v0.0.11 (November 22, 2016) | 23 |
| 11.2 | v0.0.10 (November 18, 2016) | 24 |
| 11.3 | v0.0.9 (August 23, 2016) | 24 |
| 11.4 | v0.0.8 (Mai 2, 2016) | 24 |
| 11.5 | v0.0.7 (October 20, 2015) | 25 |
| 12 | Indices and tables | 27 |
| | Python Module Index | 29 |
| | Index | 31 |

Connect weather data interfaces with interfaces of wind and pv power models.

- Free software: MIT license

1.1 Installation

On Linux systems, you can just:

```
pip install feedinlib
```

You can also install the in-development version with:

```
pip install https://github.com/oemof/feedinlib/archive/master.zip
```

On Windows systems, some dependencies are not pip-installable. Thus, Windows users first have to manually install the dependencies e.g. using conda or mamba.

1.2 Documentation

<https://feedinlib.readthedocs.io/>

1.3 Development

To run all the tests run:

```
tox
```

Note, to combine the coverage data from all the tox environments run:

| | |
|---------|--|
| Windows | <pre>set PYTEST_ADDOPTS=--cov-append tox</pre> |
| Other | <pre>PYTEST_ADDOPTS=--cov-append tox</pre> |

The feedinlib is designed to calculate feed-in time series of photovoltaic and wind power plants. It is part of the oemof group but works as a standalone application.

The feedinlib is ready to use but it definitely has a lot of space for further development, new and improved models and nice features.

2.1 Introduction

So far the feedinlib provides interfaces to download *open_FRED* and *ERA5* weather data. *open_FRED* is a local reanalysis weather data set that provides weather data for Germany (and bounding box). *ERA5* is a global reanalysis weather data set that provides weather data for the whole world. The weather data can be used to calculate the electrical output of PV and wind power plants. At the moment the feedinlib provides interfaces to the *pvl* and the *windpowerlib*. Furthermore, technical parameters for many PV modules and inverters, as well as wind turbines, are made available and can be easily used for calculations.

2.2 Installation

If you have a working Python 3 environment, use pip to install the latest feedinlib version:

```
pip install feedinlib
```

The feedinlib is designed for Python 3 and tested on Python ≥ 3.6 .

We highly recommend to use virtual environments.

2.3 Examples and basic usage

The basic usage of the feedinlib is shown in the *Examples* section. The examples are provided as jupyter notebooks that you can download here:

- ERA5 weather data example
- open_FRED weather data example
- pvlib model example
- windpowerlib model example

Furthermore, you have to install the feedinlib with additional packages needed to run the notebooks, e.g. *jupyter*.

```
pip install feedinlib[examples]
```

To launch jupyter notebook type `jupyter notebook` in the terminal. This will open a browser window. Navigate to the directory containing the notebook(s) to open it. See the jupyter notebook quick start guide for more information on [how to run jupyter notebooks](#).

2.4 Contributing

We are warmly welcoming all who want to contribute to the feedinlib. If you are interested do not hesitate to contact us via [github](#).

As the feedinlib started with contributors from the [oemof developer group](#) we use the same [developer rules](#).

How to create a pull request:

- [Fork](#) the feedinlib repository to your own github account.
- Create a local clone of your fork and install the cloned repository using pip with `-e` option:

```
pip install -e /path/to/the/repository
```

- Change, add or remove code.
- Commit your changes.
- Create a [pull request](#) and describe what you will do and why.
- Wait for approval.

Generally the following steps are required when changing, adding or removing code:

- Add new tests if you have written new functions/classes.
- Add/change the documentation (new feature, API changes ...).
- Add a whatsnew entry and your name to Contributors.
- Check if all tests still work by simply executing `pytest` in your feedinlib directory:

```
pytest
```

2.5 Citing the feedinlib

We use the zenodo project to get a DOI for each version. [Search zenodo](#) for the right citation of your feedinlib version.

2.6 License

MIT License

Copyright (C) 2017 oemof developer group

CHAPTER 3

Installation

At the command line:

```
pip install feedinlib
```


CHAPTER 4

Usage

To use feedinlib in a project:

```
import feedinlib
```


CHAPTER 5

Examples

6.1 Power plant classes

Power plant classes for specific weather dependent renewable energy resources.

| | |
|--|--|
| <code>feedinlib.powerplants.Photovoltaic([model])</code> | Class to define a standard set of PV system attributes. |
| <code>feedinlib.powerplants.WindPowerPlant([model])</code> | Class to define a standard set of wind power plant attributes. |

6.2 Feed-in models

Feed-in models take in power plant and weather data to calculate power plant feed-in. So far models using the python libraries `pvlib` and `windpowerlib` to calculate photovoltaic and wind power feed-in, respectively, have been implemented.

| | |
|---|--|
| <code>feedinlib.models.Pvlib(**kwargs)</code> | Model to determine the feed-in of a photovoltaic module using the <code>pvlib</code> . |
| <code>feedinlib.models.WindpowerlibTurbine(**kwargs)</code> | Model to determine the feed-in of a wind turbine using the <code>windpowerlib</code> . |
| <code>feedinlib.models.WindpowerlibTurbineCluster(...)</code> | Model to determine the feed-in of a wind turbine cluster using the <code>windpowerlib</code> . |

6.3 Weather data

The `feedinlib` enables download of `open_FRED` weather data (local reanalysis data for Germany) and ERA5 weather data (global reanalysis data for the whole world).

| | |
|--|---|
| <code>feedinlib.open_FRED.Weather</code> | |
| <code>feedinlib.era5.weather_df_from_era5(...[, ...])</code> | Gets ERA5 weather data from netcdf file and converts it to a pandas dataframe as required by the specified lib. |
| <code>feedinlib.era5.get_era5_data_from_dates(...)</code> | Download a netCDF file from the era5 weather data server for you position and time range. |

6.4 Tools

| | |
|---|--|
| <code>feedinlib.models.get_power_plant_data(...)</code> | Function to retrieve power plant data sets provided by feed-in models. |
|---|--|

6.5 Abstract classes

The feedinlib uses abstract classes for power plant and feed-in models that serve as blueprints for classes that implement those models. This ensures that new models provide required implementations that make it possible to easily exchange the model used in your calculation. They are important for people who want to implement new power plant and model classes rather than for users.

| | |
|---|---|
| <code>feedinlib.powerplants.Base(**attributes)</code> | The base class of feedinlib power plants. |
| <code>feedinlib.models.Base</code> | |
| <code>feedinlib.models.PhotovoltaicModelBase</code> | |
| <code>feedinlib.models.WindpowerModelBase</code> | |

7.1 feedinlib

| feedinlib | | windpowerlib | | pvlib | |
|--|------------------|------------------|------|------------|------------------|
| parameter | unit | parameter | unit | parameter | unit |
| wind_speed | m/s | wind_speed | m/s | wind_speed | m/s |
| air_temperature | K | temperature | K | temp_air | C |
| pressure | Pa | pressure | Pa | | |
| roughness_length | m | roughness_length | m | | |
| surface_normalized_global_downwelling_shortwave_flux | W/m ² | | | ghi | W/m ² |
| surface_diffuse_downwelling_shortwave_flux | W/m ² | | | dhi | W/m ² |
| surface_normalized_direct_downwelling_shortwave_flux | W/m ² | | | dni | W/m ² |

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

8.1 Bug reports

When [reporting a bug](#) please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

8.2 Documentation improvements

feedinlib could always use more documentation, whether as part of the official feedinlib docs, in docstrings, or even on the web in blog posts, articles, and such.

8.3 Feature requests and feedback

The best way to send feedback is to file an issue at <https://github.com/oemof/feedinlib/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that code contributions are welcome :)

8.4 Development

To set up *feedinlib* for local development:

1. Fork *feedinlib* (look for the “Fork” button).
2. Clone your fork locally:

```
git clone git@github.com:YOURGITHUBNAME/feedinlib.git
```

3. Create a branch for local development:

```
git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

4. When you’re done making changes run all the checks and docs builder with `tox` one command:

```
tox
```

5. Commit your changes and push your branch to GitHub:

```
git add .  
git commit -m "Your detailed description of your changes."  
git push origin name-of-your-bugfix-or-feature
```

6. Submit a pull request through the GitHub website.

8.4.1 Pull Request Guidelines

If you need some code review or feedback while you’re developing the code just make the pull request.

For merging, you should:

1. Include passing tests (run `tox`).
2. Update documentation when there’s new API, functionality etc.
3. Add a note to `CHANGELOG.rst` about the changes.
4. Add yourself to `AUTHORS.rst`.

8.4.2 Tips

To run a subset of tests:

```
tox -e envname -- pytest -k test_myfeature
```

To run all the test environments in *parallel*:

```
tox -p auto
```

CHAPTER 9

Authors

oemof developer group - <https://oemof.org>

(alphabetic order)

- Birgit Schachler
- Cord Kaldemeyer
- Francesco Witte
- gplssm
- Patrik Schönfeldt
- Pierre Francois
- Sabine Haas
- Stephan Günther
- Stephen Bosch
- Uwe Krien

CHAPTER 10

Changelog

10.1 0.0.0 (2021-06-10)

- First release on PyPI.

These are new features and improvements of note in each release

Releases

- *v0.0.11 (November 22, 2016)*
- *v0.0.10 (November 18, 2016)*
- *v0.0.9 (August 23, 2016)*
- *v0.0.8 (Mai 2, 2016)*
- *v0.0.7 (October 20, 2015)*

11.1 v0.0.11 (November 22, 2016)

11.1.1 New features

- Using model of windpowerlib instead of internal model. This will be the future of the feedinlib.

11.1.2 Bug fixes

- removed 'vernetzen'-server because it is down

11.1.3 Contributors

- Uwe Krien

11.2 v0.0.10 (November 18, 2016)

11.2.1 Other changes

Move wind power calculations to windpowerlib Allow installation of windpowerlib for python versions >3.4 Import requests package instead of urllib5

11.2.2 Contributors

- Uwe Krien
- Stephen Bosch
- Birgit Schachler

11.3 v0.0.9 (August 23, 2016)

11.3.1 Bug fixes

- Adapt API due to changes in the pvlib
- Avoid pandas future warning running the pv model

11.3.2 Contributors

- Uwe Krien

11.4 v0.0.8 (Mai 2, 2016)

11.4.1 New features

- add a geometry attribute for shapely.geometry objects to the weather class
- add lookup table for the sandia pv modules

11.4.2 Documentation

- add link to the developer rules of oemof

11.4.3 Bug fixes

- Adapt url to sandia's module library

11.4.4 Contributors

- Uwe Krien

11.5 v0.0.7 (October 20, 2015)

11.5.1 New features

- add a weather class to define the structure of the weather data input
- add example file to pass your own model class to the feedinlib

11.5.2 Documentation

- correct some typos
- some distributions are clearer now
- describe the used units

11.5.3 Testing

- add more doctests
- removed obsolete tests

11.5.4 Bug fixes

- does not overwrite class attributes (issue 7)

11.5.5 Other changes

- rename classes to more describing names
- initialisation of a power plant changed (see README for details)

11.5.6 Contributors

- Uwe Krien
- Stephan Günther
- Cord Kaldemeyer

CHAPTER 12

Indices and tables

- `genindex`
- `modindex`
- `search`

f

feedinlib, 15

F

feedinlib (*module*), 15